GNU Data Language (GDL) - a free and open-source implementation of IDL[®]

Sylwester Arabas[1], Marc Schellens[*], Alain Coulais[2], Joel Gales[3], Peter Messmer[4], and many other volunteers all around the world[**]

[*]: **Marc is the primary author and the maintainer of GDL** [**]: a list of contributors is available at the project's website (http://gnudatalanguage.sf.net/) and in the AUTHORS file of the GDL tarballs and packages
[1]: Institute of Geophysics, University of Warsaw, Poland; [2]: LERMA, CNRS and Observatoire de Paris, Paris, France; [3]: NASA Goddard Space Flight Center, Greenbelt, Maryland, USA; [4]: Tech-X Corporation, Boulder, Colorado, USA

About GDL

GNU Data Language (GDL) is developed with the aim of providing an open-source drop-in replacement for the ITTVIS's Interactive Data Language (IDL). It is free software developed by an international team of volunteers led by Marc Schellens - the project's founder (a list of contributors is available on the project's website). GDL with its library routines is designed as a tool for numerical data analysis and visualisation. As its proprietary counterparts (IDL and PV-WAVE), GDL is used particularly in geosciences and astronomy. GDL is dynamically-typed, vectorized and has object-oriented programming capabilities. The library routines handle numerical calculations, data visualisation, signal/image processing, interaction with host OS and data input/output. GDL supports several data formats such as netCDF, HDF4, HDF5, GRIB, PNG, TIFF, DICOM, etc. Graphical output is handled by X11, PostScript, SVG or z-buffer terminals, the last one allowing output to be saved in a variety of raster graphics formats. While still being in its beta-stage of development (see Coulais et al., 2009), GDL proved to be a usable tool both for research (Jaffey et al., 2008, 2009) and teaching purposes (outlined hereby). GDL is released under the GNU GPL licence - you are free to use it, to share it with anyone, to modify and improve it, and to share the modified version with anyone. Contributions and users' feedback are welcome!

References

Coulais, A., M. Schellens, J. Gales, S. Arabas, M. Boquien, P. Chanial, P. Messmer, D. Fillmore, O. Poplawski, S. Maret, G. Marchal, N. Galmiche, and T. Mermet, 2009: Status of GDL - GNU Data Language. *Astronomical Data Analysis Software and Systems XIX*, ASP Conference Series, Sapporo, Japan. http://aramis2.obspm.fr/coulais/.
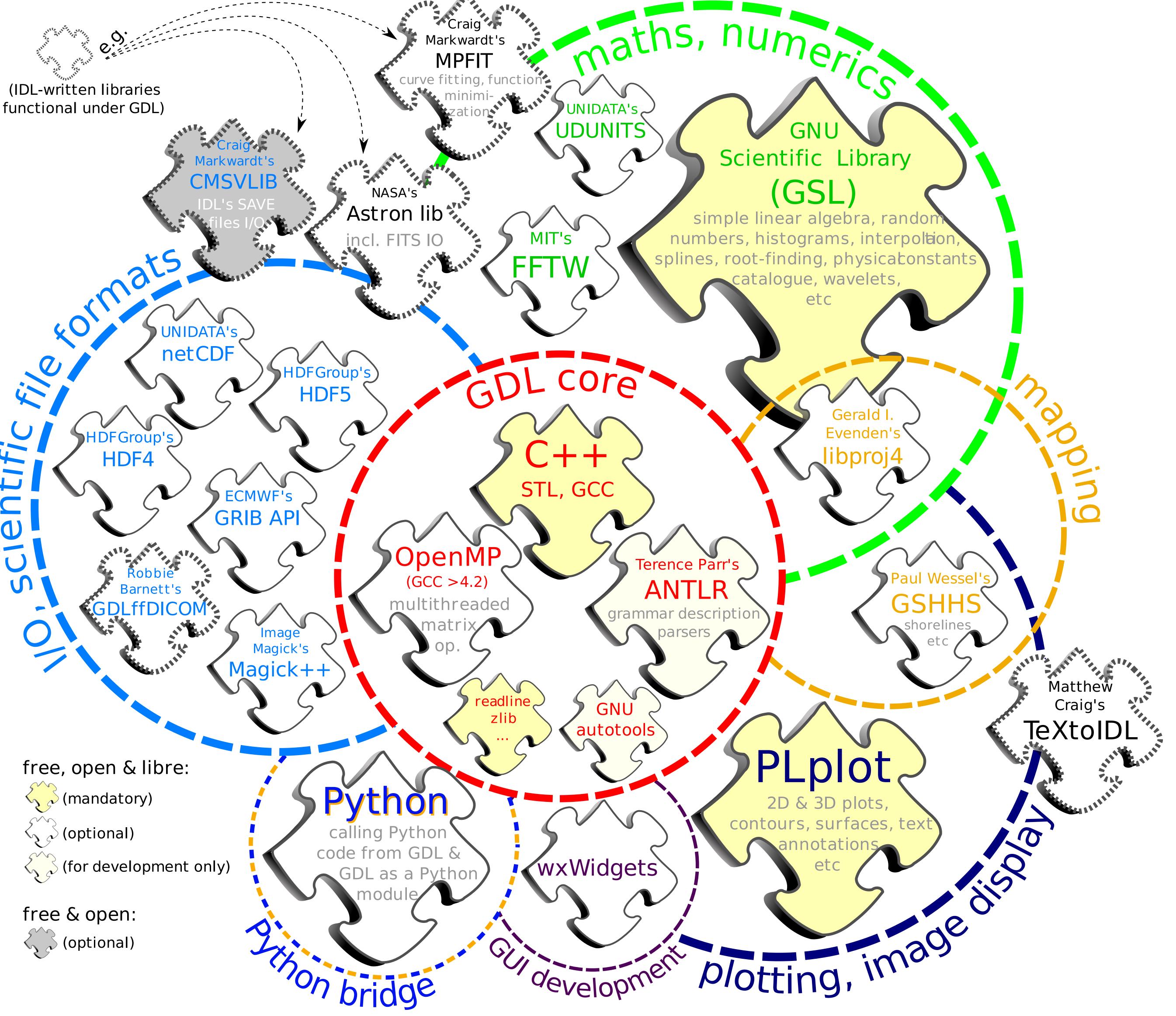
Jaffey, A., M. Cheung, and A. Kobashi, 2008: Online data analysis using Web GDL. *AGU Fall Meeting 2008*, San Francisco, USA. http://www.agu.org/meetings/fm08/.

Jaffey, A., A. Kobashi, and M. Cheung, 2009: GPU acceleration of the scientific data analysis package GDL. *NVIDIA Research Summit 2009*, San Jose, USA. http://www.nvidia.com/object/gtc 2009 archive.html.
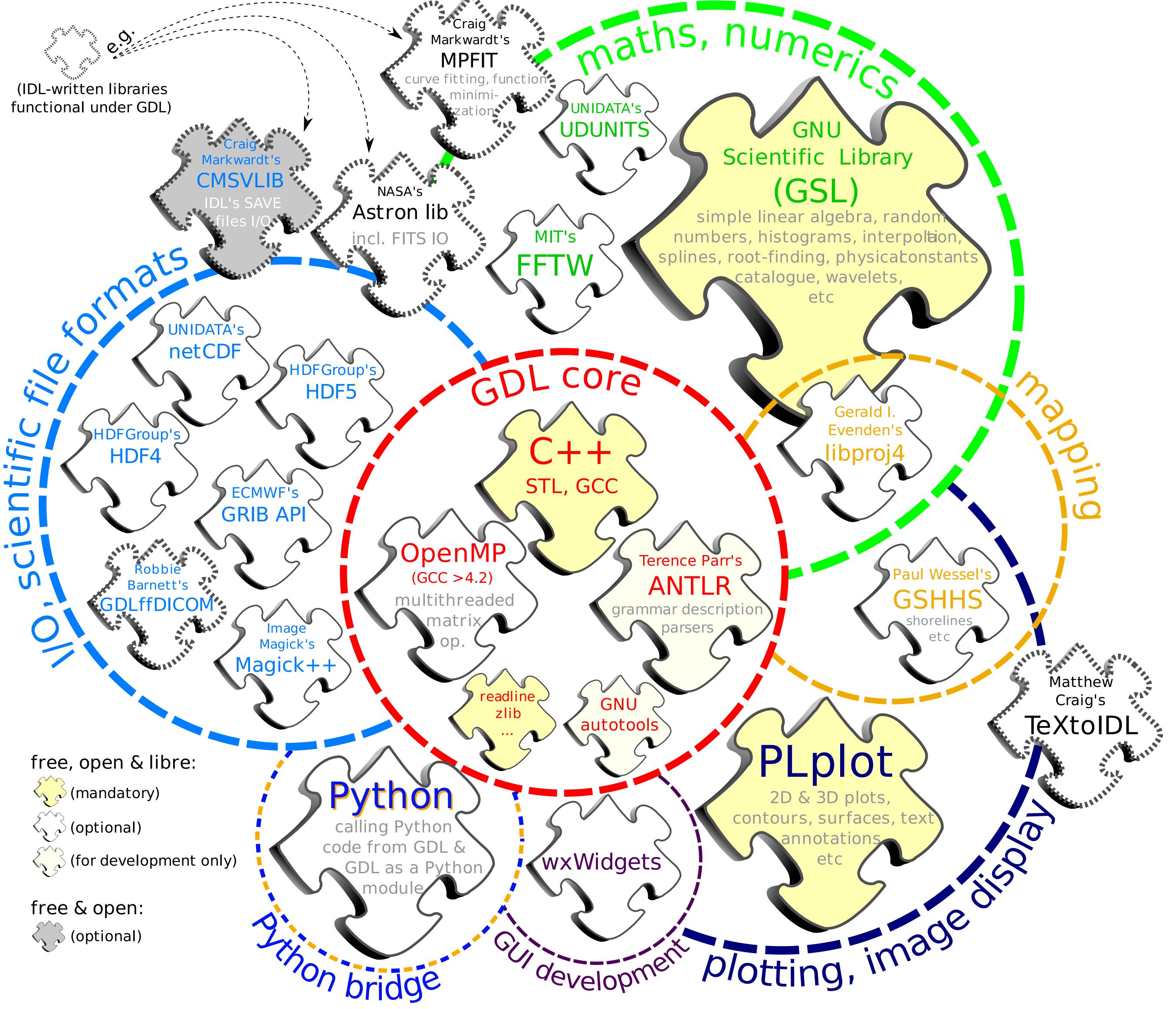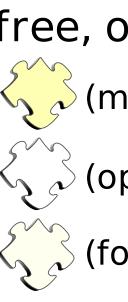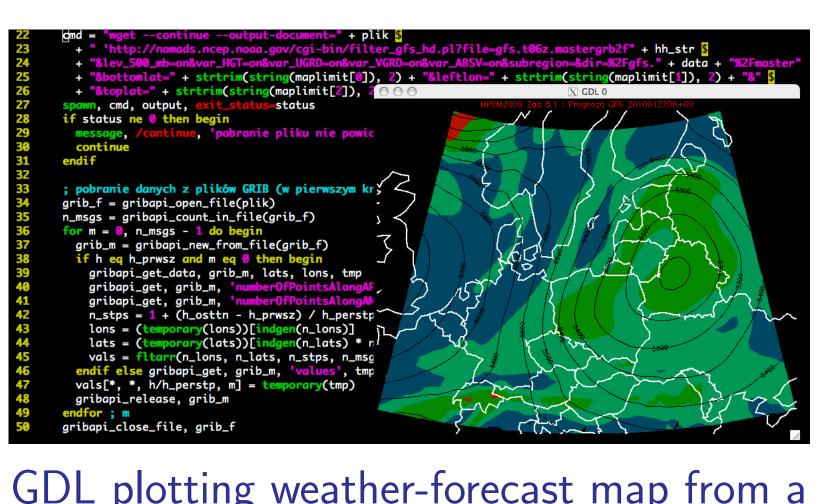
Acknowledgements

Disclaimer

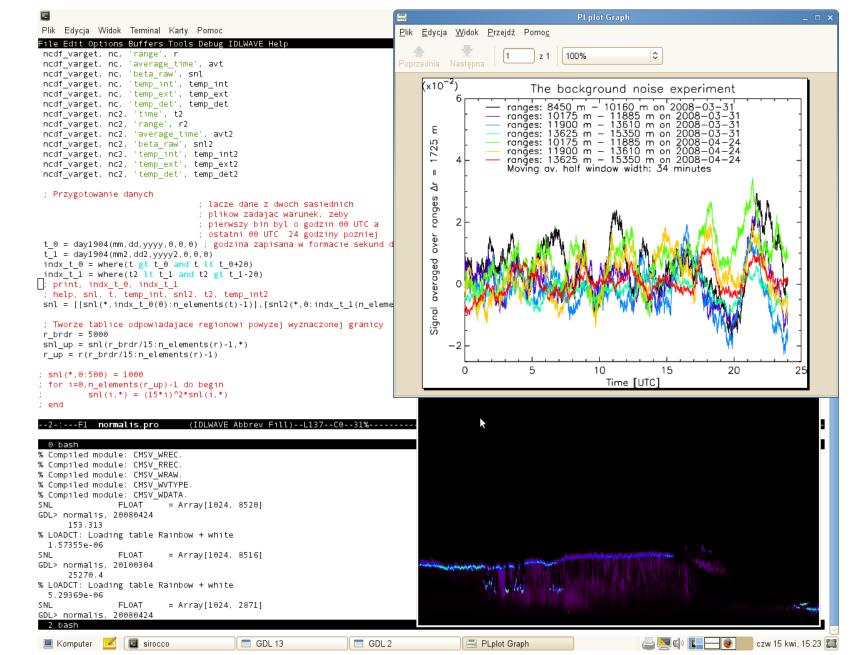## Dependencies and useful libraries (all free & open)

GDL is an incremental compiler with integrated debugging facilities. It is written in C++ using the ANTLR language-recognition framework. Extensions to GDL can be written in C++, GDL, and Python. Most of the library routines are implemented as interfaces to open-source libraries:

Example usage of GDL for classroom work on data analysis

GDL proved to be a useful tool for classroom work on data analysis. In 2010 it was used for the second time as a data-analysis environment during a meteorological-data processing course taught at the Institute of Geophysics, University of Warsaw. The course covers hands-on training in handling of scientific data sets. GDL is used for reading GRIB, netCDF & HDF files, and performing basic data analysis and visualisation tasks.

GDL compressing weather-radar images using truncated wavelet approximation.

GDL plotting weather-forecast map from a GRIB file with NOAA GFS model output.

GDL used as a part of a web-interface for controlling 2D fluid flow simulation and visualizing the output (reading netCDF files and outputting SVG plots).

GDL used for demonstrating Kalman filtering on randomly-generated signal.

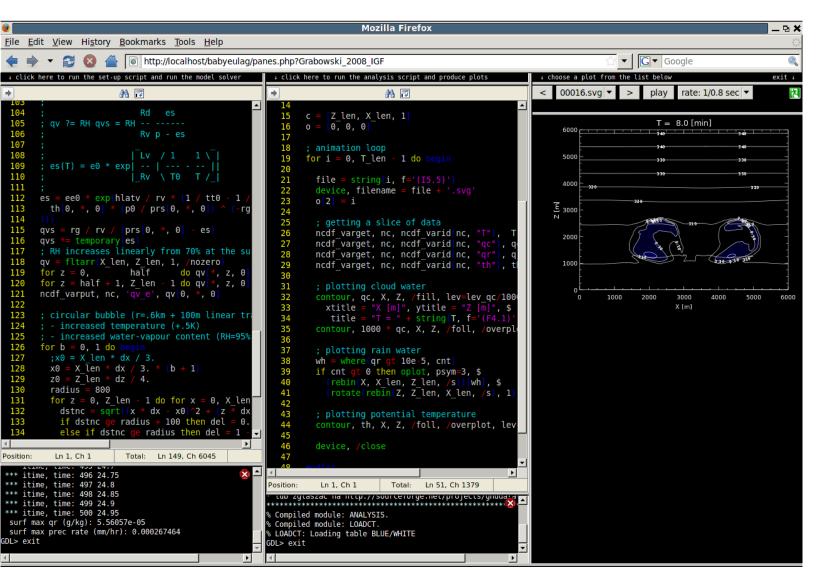GDL used for analyzing LIDAR data (reading netCDF files, producing PostScript plots) on Linux, using the IDLWAVE mode for Emacs.

GDL-generated PostScript plot of a Fourier spectrum in a LaTeX-generated PDF file.

GDL displaying MODIS satellite image (stored in a HDF file) on multiple X11 windows - using a true-color RGB composite, and using separate channels.
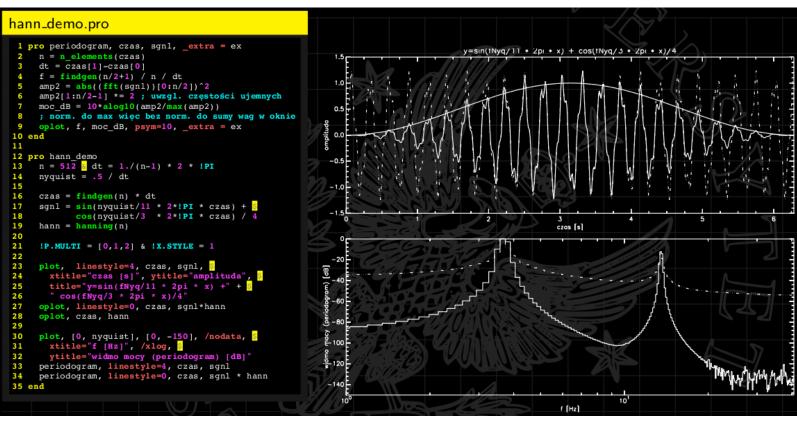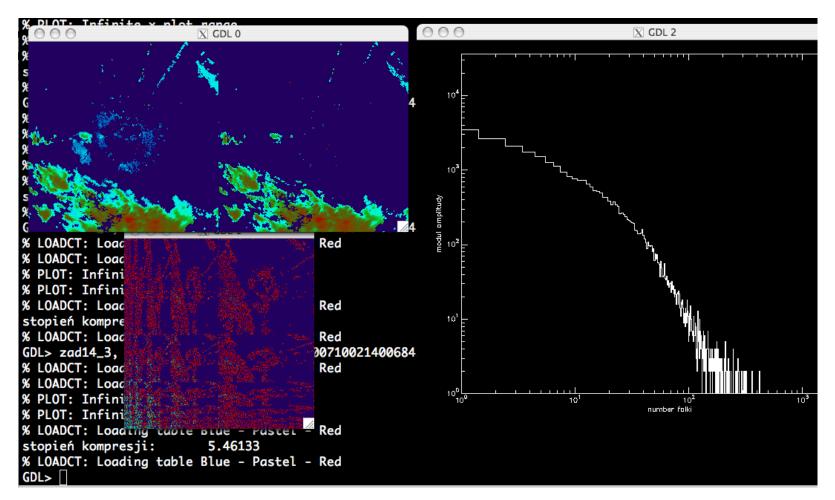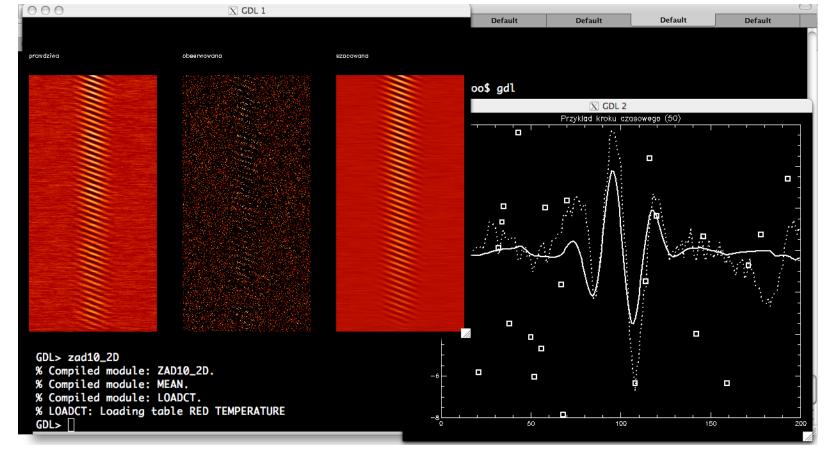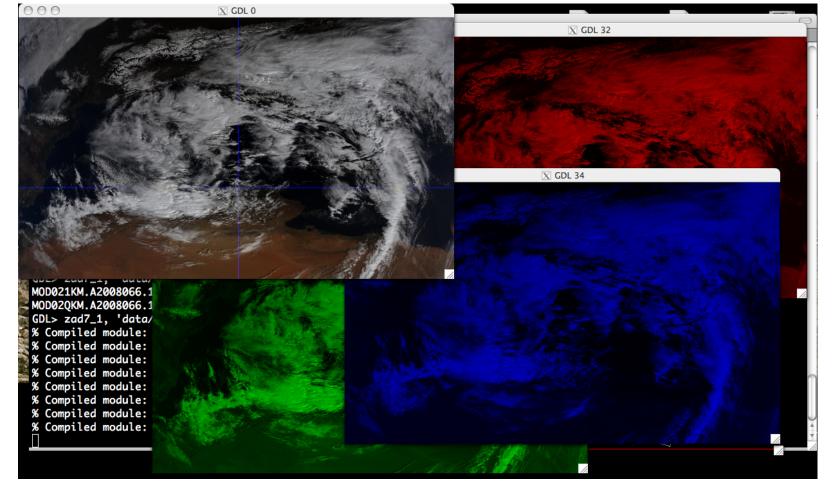
Supported platforms and availability of pre-compiled/pre-configured packages

Packaged versions of GDL are available for several Linux distributions (incl. Fedora, Gentoo and Debian) and Mac OS X (e.g. via Macports and Fink). The source code (available at http://gnudatalanguage.sf.net/) compiles on other UNIX systems, including BSD, OpenSolaris.